

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Currently Amended) A method for moving data objects in a computer system from a first storage location to a second storage location, the method comprising:

selecting a data object from the first storage location;

assigning an identifier (ID) to the data object;

storing the ID in a transactional type lock object;

determining whether the ID is stored successfully in the transactional type lock object, and upon a successful storage, storing the ID in a permanent type lock object, thereby indicating that the data object is stored at the first storage location;

determining whether the ID is stored successfully in the permanent type lock object, and upon a successful storage, deleting the ID from the transactional type lock object;

storing the data object at the second storage location;

assigning the second storage location to the ID in the permanent type lock object;

deleting the data object from the first storage location; and

deleting the ID from the permanent type lock object, thereby indicating that the data object is not stored at the first storage location, after the ~~respective~~ data object assigned to that ID has been deleted from the first storage location.

2. (Previously Presented) The method of claim 1, wherein the data object comprises one or more fields of one or more tables, and wherein the ID comprises one or more key fields of the one or more tables.

3. (Currently Amended) The method of claim 1, wherein the data object is stored in a file and wherein an assignment of the ID to the file or a name of the file, ~~in which the data object assigned to the ID is to be stored~~, is stored in the permanent type lock object.

4. (Previously Presented) The method of claim 1, wherein the ID is stored in the transactional type lock object after assigning the ID to the data object.

5. (Currently Amended) The method of claim 1, wherein storing the ID in the permanent type lock object comprises storing IDs of ~~all selected data~~ other objects in the permanent type lock object before storing ~~any of the selected data~~ object objects at the second storage location.

6. (Currently Amended) The method of claim 1, further comprising:

checking whether the ID ~~of the data object~~ has been stored in at least one of the transactional type lock object and the permanent type lock object, and if the ID has been stored, skipping storing the data object at the second storage location.

7. (Previously Presented) The method of claim 1, further comprising:

checking whether the data object is contained in the second storage location and if the data object is contained, skipping storing the data object at the second storage location.

8. (Previously Presented) The method of claim 7, wherein the checking comprises querying at least one of the transactional type lock object and the permanent type lock object.

9. (Currently Amended) The method of claim 1, further comprising:

determining whether the ID was successfully stored in the transactional type lock object, and upon an unsuccessful storage, checking whether the data object assigned ~~to the respective ID~~ has been stored in the second storage location, and if the data object has not been stored, skipping deleting the data object from the first storage location and skipping deleting the ID from the permanent type lock object.

10. (Previously Presented) The method of claim 1 for use in an enterprise resource planning software.

11. (Currently Amended) A computer system for processing data, the computer system comprising:

memory means for storing program instructions;

input means for entering the data;

storage means for storing the data;

a processor responsive to the program instructions, wherein the program instructions comprise program code means for performing a method for moving data objects in the computer system from a first storage location to a second storage location, the method comprising:

selecting a data object from the first storage location;

assigning an identifier (ID) to the data object;

storing the ID in a transactional type lock object;

determining whether the ID is stored successfully in the transactional type lock object, and upon a successful storage, storing the ID in a permanent type lock object, thereby indicating that the data object is stored at the first storage location;

determining whether the ID is stored successfully in the permanent type lock object, and upon a successful storage, deleting the ID from the transactional type lock object;

storing the data object at the second storage location;

assigning the second storage location to the ID in the permanent type lock object;

deleting the data object from the first storage location; and

deleting the ID from the permanent type lock object, thereby indicating that the data object is not stored at the first storage location, after the ~~respective~~ data object assigned to that ID has been deleted from the first storage location.

12. (Currently Amended) A computer readable storage medium comprising instructions for performing a method for moving data objects in a computer system from a first storage location to a second storage location, the method comprising:

selecting a data object from the first storage location;

assigning an identifier (ID) to the data object;

storing the ID in a transactional type lock object;

determining whether the ID is stored successfully in the transactional type lock object, and upon a successful storage, storing the ID in a permanent type lock object, thereby indicating that the data object is stored at the first storage location;

determining whether the ID is stored successfully in the permanent type lock object, and upon a successful storage, deleting the ID from the transactional type lock object;

storing the data object at the second storage location;

assigning the second storage location to the ID in the permanent type lock object;

deleting the data object from the first storage location; and

deleting the ID from the permanent type lock object, thereby indicating that the data object is not stored at the first storage location, after the respective data object ~~assigned to that ID~~ has been deleted from the first storage location.

13-14. (Canceled).

15. (Currently Amended) The computer readable storage medium of claim 12, wherein the data object comprises one or more fields of one or more tables, and wherein the ID comprises one or more key fields of the one or more tables.

16. (Currently Amended) The computer readable storage medium of claim 12, wherein the data object is stored in a file and wherein an assignment of the ID to the file or a name of the file, ~~in which the data object assigned to the ID is to be stored~~, is stored in the permanent type lock object.

17. (Currently Amended) The computer readable storage medium of claim 12, wherein the ID is stored in the transactional type lock object after assigning the ID to the data object.

18. (Currently Amended) The computer readable storage medium of claim 12, wherein storing the ID in the permanent type lock object comprises storing IDs of ~~all selected data~~ other objects in the permanent type lock object before storing ~~any of the selected data~~ object ~~objects~~ at the second storage location.

19. (Currently Amended) The computer readable storage medium of claim 12, wherein the method further comprises:

checking whether the ID ~~of the data object~~ has been stored in at least one of the transactional type lock object and the permanent type lock object, and if the ID has been stored, skipping storing the data object at the second storage location.

20. (Currently Amended) The computer readable storage medium of claim 12, wherein the method further comprises:

checking whether the data object is contained in the second storage location and if the data object is contained, skipping storing the data object at the second storage location.

21. (Currently Amended) The computer readable storage medium of claim 20, wherein the checking comprises querying at least one of the transactional type lock object and the permanent type lock object.

22. (Currently Amended) The computer readable storage medium of claim 12, wherein the method further comprises:

determining whether the ID was successfully stored in the transactional type lock object, and upon an unsuccessful storage, checking whether the data object assigned to the respective ID has been stored in the second storage location, and if the data object has not been stored, skipping deleting the data object from the first storage location and skipping deleting the ID from the permanent type lock object.

23. (Currently Amended) A computer system for processing data, the computer system comprising:

a processor executing program instructions;

means for assigning an identifier (ID) to the data object;

means for storing the ID in a transactional type lock object;

means for determining whether the ID is stored successfully in the transactional type lock object, and upon a successful storage, storing the ID in a permanent type lock object, thereby indicating that the data object is stored at the first storage location;

means for determining whether the ID is stored successfully in the permanent type lock object, and upon a successful storage, deleting the ID from the transactional type lock object;

means for storing the data object at the second storage location;

means for assigning the second storage location to the ID in the permanent type lock object;

means for deleting the data object from the first storage location; and

means for deleting the ID from the permanent type lock object, thereby indicating that the data object is stored at the first storage location, after the respective data object assigned to that ID has been deleted from the first storage location.

24. (Previously Presented) The computer system of claim 23, wherein the data object comprises one or more fields of one or more tables, and wherein the ID comprises one or more key fields of the one or more tables.

25. (Currently Amended) The computer system of claim 23, further comprising:

means for storing the data object in a file; and

means for storing an assignment of the ID to the file or a name of the file, ~~in which the data object assigned to the ID is to be stored~~, in the permanent type lock object.

26. (Previously Presented) The computer system of claim 23, further comprising means for storing the ID in the transactional type lock object after assigning the ID to the data object.

27. (Currently Amended) The computer system of claim 23, wherein the means for storing the ID in the permanent type lock object comprises means for storing IDs of ~~all selected data~~ other objects in the permanent type lock object before storing ~~any of the selected data~~ object ~~objects~~ at the second storage location.

28. (Previously Presented) The computer system of claim 23, further comprising:

means for checking whether the ID of the data object has been stored in at least one of the transactional type lock object and the permanent type lock object, and if the ID has been stored, skipping storing the data object at the second storage location.

29. (Previously Presented) The computer system of claim 23, further comprising:

means for checking whether the data object is contained in the second storage location and if the data object is contained, skipping storing the data object at the second storage location.

30. (Previously Presented) The computer system of claim 29, wherein the means for checking comprises means for querying at least one of the transactional type lock object and the permanent type lock object.

31. (Currently Amended) The computer system of claim 23, further comprising:

means for determining whether the ID was successfully stored in the transactional type lock object, and upon an unsuccessful storage, checking whether the data object ~~assigned to the respective ID~~ has been stored in the second storage location, and if the data object has not been stored, skipping deleting the data object from the first storage location and skipping deleting the ID from the permanent type lock object.

32. (New) The method of claim 1, wherein storing the ID in the transactional type lock object indicates that an action is being performed on the data object.

33. (New) The method of claim 32, wherein deleting the ID from the transactional type lock object indicates that the action is not being performed on the data object.

34. (New) The computer system of claim 11, wherein storing the ID in the transactional type lock object indicates that an action is being performed on the data object.

35. (New) The computer system of claim 34, wherein deleting the ID from the transactional type lock object indicates that the action is not being performed on the data object.

36. (New) The computer readable storage medium of claim 12, wherein storing the ID in the transactional type lock object indicates that an action is being performed on the data object.

37. (New) The computer readable storage medium of claim 36, wherein deleting the ID from the transactional type lock object indicates that the action is not being performed on the data object.

38. (New) The computer system of claim 23, wherein storing the ID in the transactional type lock object indicates that an action is being performed on the data object.

39. (New) The computer system of claim 38, wherein deleting the ID from the transactional type lock object indicates that the action is not being performed on the data object.